

A UTILIZAÇÃO DE ALGORITMOS NOS PROCESSOS DE ENSINO E DE APRENDIZAGEM DE PROGRAMAÇÃO DE COMPUTADORES EM CURSOS TÉCNICOS

Tiago Gossmann¹
Edson Moacir Ahlert²

Resumo: Os processos de ensino e de aprendizagem de programação de computadores sempre foram um grande desafio, tanto para professores quanto estudantes. Frequentemente registram-se dificuldades por parte dos discentes na compreensão destes assuntos o que acaba gerando desinteresse, reprovação e evasão. Neste trabalho são levantados diversos fatores que contribuem para este cenário e propõe-se alternativas para amenizar esta realidade. Este artigo relata a experiência da inserção de Algoritmos no conteúdo programático inicial da disciplina de Programação I, do curso técnico em Eletroeletrônica da Univates, cuja finalidade foi preparar os estudantes a terem melhor compreensão da programação de computadores frente ao uso de uma linguagem de desenvolvimento de alto nível, utilizada em ambientes reais de produção de *software*. Os resultados obtidos foram considerados satisfatórios, uma vez que minimizaram as dificuldades apresentadas e favoreceram a aprendizagem dos estudantes, deixando-os melhor preparados para os conteúdos que virão na sequência do curso. Além disso, a prática resultou em uma diminuição no índice de evasão da disciplina, quando comparado a semestres anteriores.

Palavras-chave: Programação de computadores. Algoritmos. Cursos Técnicos.

1 Introdução

Com a evolução da Tecnologia da Informação há muitas áreas onde a programação de computadores pode ser explorada. O que no passado era algo restrito a computadores, nos dias atuais pode ser expandida para dispositivos móveis e também para a automação industrial e residencial, por exemplo. Conforme surgem novas plataformas, necessita-se também formar profissionais tecnicamente capacitados (leia-se programadores) para trabalhar no desenvolvimento de programas que sejam executados nestes dispositivos.

Entretanto, estudantes destes conteúdos demonstram, regularmente, dificuldades na assimilação e entendimento dos assuntos relativos a programação de computadores. Trabalhar

¹Graduado em Tecnologia em Análise e Desenvolvimento de Sistemas. Acadêmico do Curso de Pós-Graduação Lato Sensu - Especialização em Docência na Educação Profissional da Univates.

²Professor da Univates. Mestre em Ambiente e Desenvolvimento.

com linguagens de programação de baixo ou alto nível, isto é, as que são utilizadas em um ambiente real de produção de *software*, necessitam um bom tempo de preparação e persistência para alcançar resultados satisfatórios.

Ensinar programação de computadores não é somente transmitir aos estudantes comandos e instruções que o computador deverá executar. Trata-se de um processo complexo, que envolve lógica, raciocínio e habilidade por parte do aluno, no conhecimento do problema a ser resolvido e a elaboração da solução para ele, muitas vezes limitando-se a uma esfera teórica.

Para amenizar essas dificuldades, torna-se necessário encontrar soluções que facilitem aos estudantes aprender de uma forma lúdica o “caminho das pedras” e, para tanto, estarem melhor preparados para trabalhar com essas linguagens de programação. Diante disso este artigo relata uma experiência com a inserção de Algoritmos no conteúdo programático da disciplina de Programação I do curso técnico em Eletroeletrônica da Univates, destacando as melhorias alcançadas em relação ao aprendizado na disciplina.

2 AS DIFICULDADES NO ENSINO E NA APRENDIZAGEM DE PROGRAMAÇÃO DE COMPUTADORES

Compete ao ensino da programação de computadores desenvolver habilidades e capacitar os estudantes a desenvolverem programas cujo objetivo é a resolução de problemas reais (SILVA; TRENTIN, 2016). Para construirmos estes programas de computador necessitamos de uma linguagem de programação real. Ao programador cabe a tarefa de escrever comandos e instruções de acordo com a linguagem de programação escolhida. Não é o objetivo deste artigo levantar a quantidade de linguagens de programação existentes nem avaliar seus pontos positivos e negativos.

Linguagens de programação costumam possuir um grande rigor técnico nas suas instruções. Isso naturalmente cria barreiras no processo de aprendizagem. Além disso a construção de um programa de computador exige estratégia e pensamento lógico-dedutivo para chegar na solução adequada (FALCKEMBACH; ARAUJO, 2006). Experiências mostram que o problema já ocorre na fase inicial da aprendizagem, quando se torna necessário aplicar certos conceitos abstratos de programação como, por exemplo, estruturas de repetição (Gomes et al, 2008).

Ainda conforme (Gomes et al, 2008) existem diversos fatores que levam ao insucesso do estudante na compreensão da programação de computadores. Os métodos de ensino e as estratégias do professor muitas vezes não são adequados a necessidade dos alunos pois exigiriam um ensino personalizado. Além disso a falta de interesse dos discentes pelo conteúdo proposto visto que exige persistência e prática, somado ao fato de não estudarem suficientemente o conteúdo para adquirir as competências necessárias dificulta ainda mais a aprendizagem. Existem ainda outros obstáculos na compreensão do tema, tais como: a enorme dificuldade na resolução de problemas, a falta de raciocínio matemático e lógico pois o assunto é correlacionado a estas áreas e a complexidade das linguagens de programação reais.

Outro fator preponderante é a quantidade de alunos por turma. No caso da disciplina em questão, temos uma média de 20 alunos por turma. Considerando que cada indivíduo possui sua maneira de compreender e assimilar o conteúdo proposto, torna-se difícil um *feedback* individual. Conforme relatam Falckembach e Araujo (2006, p. 2).

A forma de ensino dentro da sala de aula é a mesma para todos os alunos, pois, é extremamente difícil para um professor levar em consideração o perfil, as metas, as necessidades, as expectativas, as preferências e o nível de conhecimento de cada aluno, de modo a proporcionar a cada um, um ensino adaptado.

Conforme visto, percebe-se que os desafios são múltiplos e exigem uma exímia percepção do docente para contornar as barreiras da aprendizagem. Propõe-se no capítulo a seguir encontrar alternativas para amenizar estas dificuldades iniciais explorando o tema Algoritmos e apurando a Lógica de Programação.

3 A LÓGICA DE PROGRAMAÇÃO E ALGORITMOS

Quando falamos em Lógica de Programação, logo nos vem à mente palavras como raciocínio, organização, ordenação, quebra-cabeças, etc. São alguns termos que traduzem a necessidade de organizar as ideias, ordenadamente, para construção de um programa de computador. Forbellone e Eberspächer (2005, p. 3) destacam que “na medida em que precisamos especificar uma sequência de passos, é necessário utilizar ordem, ou seja, ‘pensar com ordem’, portanto precisamos utilizar lógica”.

A partir da necessidade de utilizar este raciocínio lógico é que começamos a encontrar as primeiras dificuldades e restrições para se conseguir elaborar um programa de computador. Conforme Manzano e Oliveira (2016, p. 27).

Muitos gostam de afirmar que possuem e sabem usar o raciocínio lógico, porém quando questionados direta ou indiretamente, perdem essa linha de raciocínio, pois

inúmeros fatores são necessários para completá-lo, tais como conhecimento, versatilidade, experiência, criatividade, responsabilidade, ponderação, calma, autodisciplina, entre outros. [...] O raciocínio lógico usado na prática da programação de computadores pode ser entendida como uma técnica de se encadear pensamentos com o intuito de atender a um certo objetivo. A lógica de programação objetiva o estabelecimento de uma sequência organizada de passos que serão executados por um computador, na forma de um programa.

Ao falar, precisamente, desta sequência organizada de passos estamos nos referindo ao termo Algoritmo. Conforme Manzano e Oliveira (2016, p. 25) o termo Algoritmo pode ser entendido como “regras formais, sequenciais e bem definidas a partir do entendimento lógico de um problema a ser resolvido por um programador com o objetivo de transformá-lo em um programa que seja possível de ser tratado e executado por um computador”. Forbellone e Eberspächer (2005) afirmam que construir Algoritmos coerentes e válidos é o principal objetivo da Lógica de Programação.










Um programador pode construir Algoritmos por meio de uma representação gráfica, utilizando um conjunto de símbolos que representam os passos do Algoritmo e onde cada um possui uma instrução a ser executada e também por meio de uma representação textual, isto é, uma forma genérica sem utilizar nenhuma linguagem de programação (MANZANO et al., 2015).

3.1 Diagrama de Blocos

Um diagrama de blocos é uma representação gráfica, através de símbolos, que ilustra o fluxo de operações que um Algoritmo efetua. Nele não são especificados detalhes técnicos. Conforme Manzano e Oliveira (2016, p. 28), “a técnica de uso e desenvolvimento de diagramas concede ao profissional da área de desenvolvimento facilidade na posterior codificação e também manutenção do programa em qualquer uma das linguagens formais de programação existentes”.

Por isso, cabe apresentar a Figura 1, na qual estão listados alguns símbolos utilizados no diagrama de blocos. Esses desenhos são estabelecidos através da norma internacional ISO 5807:1985 e permitem demonstrar de forma clara a linha de raciocínio lógico utilizado por um programador de computador (MANZANO; OLIVEIRA, 2016).

Figura 1 – Alguns elementos utilizados no diagrama de blocos.

Símbolo	Significado	Descrição
	Terminal <i>Terminator</i>	O símbolo representa a definição de início e fim do fluxo lógico de um programa. Também é utilizado na definição de sub-rotinas de procedimento ou de função.
	Entrada manual <i>Manual input</i>	Representa a entrada manual de dados, normalmente efetuada em um teclado conectado diretamente ao console do computador.
	Processamento <i>Process</i>	Representa a execução de uma operação ou grupo de operações que estabelecem o resultado de uma operação lógica ou matemática.
	Exibição <i>Display</i>	Representa a execução da operação de saída visual de dados em um monitor de vídeo conectado ao console do computador.
	Decisão <i>Decision</i>	O símbolo representa o uso de desvios condicionais para outros pontos do programa de acordo com situações variáveis.
	Preparação <i>Preparation</i>	Representa a modificação de instruções ou grupo de instruções existentes em relação à ação de sua atividade subsequencial.
	Processo predefinido <i>Predefined process</i>	Definição de um grupo de operações estabelecidas como uma sub-rotina de processamento anexa ao diagrama de blocos.
	Conector <i>Connector</i>	Representa a entrada ou a saída em outra parte do diagrama de blocos. Pode ser usado na definição de quebras de linha e na continuação da execução de decisões.
	Linha <i>Line</i>	O símbolo representa a ação de vínculo existente entre os vários símbolos de um diagrama de blocos. Possui a ponta de uma seta indicando a direção do fluxo de ação.

Fonte: Manzano e Oliveira (2016, p. 31).

3.2 Pseudocódigo

O termo pseudocódigo também é conhecido como português estruturado, metalinguagem ou portugol. É através dele que transcrevemos, em instruções genéricas, os comandos que um Algoritmo deve executar. O pseudocódigo não possui a rigidez que uma linguagem de programação possui, mas é um método extremamente didático de se compreender a forma como se desenvolve um programa de computador. Em outras palavras, pode-se dizer que o pseudocódigo é uma espécie de simulador de uma linguagem de programação real.

Aguilar (2011, p. 61) escreve que “o pseudocódigo é uma linguagem de especificação (descrição) de Algoritmos. O uso dessa linguagem torna a codificação final (isto é, a tradução para uma linguagem de programação) relativamente fácil.”. Complementam Goodrich e Tamassia (2008, p. 23) complementam que o pseudocódigo “é uma mistura de linguagem natural e estruturas de programação de alto nível usada para descrever as ideias principais da implementação genérica de uma estrutura de dados ou Algoritmo”.

Diante destas definições, a Tabela 1 apresenta um comparativo dos principais tipos de dados e instruções entre uma linguagem de programação de alto nível, no caso a linguagem C, e a mesma instrução em pseudocódigo. É possível observar a familiaridade que o pseudocódigo assume com a língua portuguesa. Dessa forma é possível comparar e justificar a facilidade em se trabalhar com este recurso.

Tabela 1 – Comparativo entre instruções de linguagem de alto nível com o pseudocódigo.

Instrução em Linguagem de Alto Nível	Instrução em Pseudocódigo
int	inteiro
float	real
char	caractere
printf	escreva
scanf	leia
for	para
while	enquanto
do...while	faça...enquanto
if	se
switch/case	caso

Fonte: Do autor (2017).

A Tabela 2 relaciona alguns ambientes para aprendizagem do pseudocódigo, todos disponíveis gratuitamente para *download* na Internet. Foram levantadas as principais características de cada um destes ambientes.

Tabela 2 – Levantamento de ambientes de pseudocódigo e suas principais características.

Software	Desenvolvedor	Características
VisuAlg	Apoio Informática	Disponível para ambiente Windows e necessita de instalação. Foi utilizado na disciplina mencionada neste artigo.
Scratch	MIT	Disponível <i>online</i> através do <i>site</i> e também oferece um instalador <i>desktop</i> para Windows, Linux e Mac. Seu objetivo é ensinar programação de forma lúdica explorando um cenário de jogo

		de computador. Pode ser considerado como uma “evolução do logo”.
Ambap	Universidade Federal de Alagoas (TCI/UFAL)	Baseado no ILA - Interpretador de Linguagem Algorítmica desenvolvido pela Unisinos-RS. Possui uma versão <i>desktop</i> desenvolvido em Java. Em sua página na internet, ao que consta, não tem mais recebido atualizações.
ILA (Interpretador de Linguagem Algorítmica)	Unisinos	Ambiente desenvolvido para MS-DOS, muito útil no aprendizado de Lógica de Programação. Possui várias funções. Seu sucessor é o AM-BAP.
Portugol Studio	Universidade do Vale do Itajaí - UNIVALI	Disponível para ambiente Windows. Ferramenta bastante moderna, totalmente remodelada, conta com vários exemplos e possui técnicas de orientação a objetos.
SuperLogo	Núcleo de Informática Aplicada à Educação (NIED-Unicamp)	Ferramenta clássica que possui inúmeras versões de vários desenvolvedores, é utilizado normalmente com crianças para aprendizagem de lógica de programação.

Fonte: Do autor (2017).

4 O USO DE ALGORITMOS

Apresenta-se, na Tabela 3, a descrição da metodologia utilizada neste trabalho, ilustrando o caminho percorrido desde a escolha do assunto, coleta das informações até a produção do referencial teórico, objetivando confirmar e buscar solucionar o problema enfrentado.

Tabela 3 – Descrição da metodologia utilizada.

Quanto à natureza	Aplicada	Gerar conhecimentos para aplicações práticas destinadas à solução de problemas específicos.
Quanto à abordagem ao problema	Qualitativa	A pesquisa relata a experiência da aplicação de Algoritmos no conteúdo da disciplina de Programação I e expõe os principais resultados obtidos.
Quanto aos objetivos	Exploratória	Analisa o desempenho dos estudantes frente a inserção de Algoritmos no conteúdo programático.
Quanto aos procedimentos técnicos	Experimental	Busca justificar a importância da utilização de Algoritmos para auxiliar os estudantes na compreensão da programação de computadores.

Fonte: Do autor (2017).

Com base na experiência docente, identificou-se que os estudantes da disciplina de Programação I do curso técnico em Eletroeletrônica da Univates apresentavam dificuldades em iniciar o desenvolvimento de programas de computador utilizando linguagem de programação de alto nível.

Buscando maiores informações sobre este cenário, por meio de conversas informais com os próprios discentes e também pela pesquisa em artigos e bibliografias, constatou-se que existe um consenso generalizado sobre essa dificuldade na iniciação do aprendizado de linguagens de programação.

Observou-se também que em determinados momentos havia um ar de desânimo por parte dos alunos, dado a dificuldade enfrentada nos conteúdos programáticos - talvez por muitos não terem um contato direto com sistemas computacionais no cotidiano.

Comparado a outras disciplinas do mesmo curso, observa-se uma quantidade considerável de reprovações e até evasão do curso. A principal queixa dos alunos era de que o conteúdo não estava sendo compreendido satisfatoriamente.

Para agravar o problema, detectaram-se comentários no sentido de o professor não conseguir sanar totalmente as dúvidas dos estudantes, por exemplo quando algum programa não funcionava corretamente, visto que o número de alunos é considerável para ser atendido individualmente.

Em contrapartida, certos estudantes registram facilidades em assimilar o conteúdo proposto. Notou-se que alguns colegas possuem um raciocínio lógico bem mais apurado comparado a outros, o que facilita consideravelmente o aprendizado de programação, entretanto, também gera um relevante desnivelamento de conhecimentos na turma. Estes estudantes, por sua vez, também desempenham um papel de auxiliares para com os colegas que possuem maior dificuldade, apoiando o professor na tentativa de sanar as dúvidas geradas pelo grupo.

A disciplina de Programação I, cujo objetivo é formar profissionais capacitados a desenvolver programas de computador, possui uma carga horária de 60 horas e está enquadrada no terceiro semestre do curso técnico em Eletroeletrônica da Univates. A média dos alunos por turma é de 20 estudantes e a linguagem de programação utilizada é a linguagem C, amplamente difundida e utilizada mundialmente.

A realidade da sala de aula nos cursos técnicos é bastante diversificada. Observam-se componentes que podem contribuir para um cenário desigual e que interfere diretamente no processo de aprendizagem. Variáveis como idade do indivíduo, se o trabalho está ligado a área de estudo ou não, tempo e distância de deslocamento, cansaço (visto que o curso é noturno e a maioria dos alunos trabalha durante o dia), etc.

A partir destas informações, buscaram-se alternativas na tentativa de amenizar estas dificuldades. O objetivo principal foi o de auxiliar o aluno a melhorar o processo de aprendizagem no desenvolvimento de programas de computador, buscando gradualmente um nivelamento de conhecimento entre os colegas, evitando também reprovação e evasão na disciplina e curso.

Esta análise não se aprofundou em uma avaliação mais criteriosa das características da turma, limitando-se a levantar informações parciais sobre ela. No caso dos estudantes foi realizado um levantamento numérico das opiniões de cada um. O objetivo é compreender as dificuldades iniciais que a maioria deles tem na disciplina, normalmente por faltar conhecimentos de lógica, mas também outros fatores, como falta de contato com tecnologia (especificamente computadores), por exemplo.

Com o intuito de não comprometer a ementa da disciplina, inseriu-se nas cinco primeiras aulas o tema Algoritmos e o conteúdo programático destas foi rearranjado para as aulas posteriores, não comprometendo assim o andamento das atividades.

Partindo do pressuposto que o estudante já conhecia o funcionamento básico de um sistema computacional, conteúdo trabalhado em disciplinas de semestres anteriores, abordou-se primeiramente a necessidade da elaboração de um Algoritmo para solução de problemas do cotidiano.

Aguilar (2011, p. 37) defende que a “primeira fase da resolução de um problema com computador é a análise do problema. Ela requer uma clara definição, na qual sejam observados exatamente o que deve fazer o programa e o resultado ou solução desejada”. A Figura 2, a seguir, ilustra os passos para resolução de um problema utilizando o computador.

Figura 2 – Passos para resolução de um problema utilizando computador.



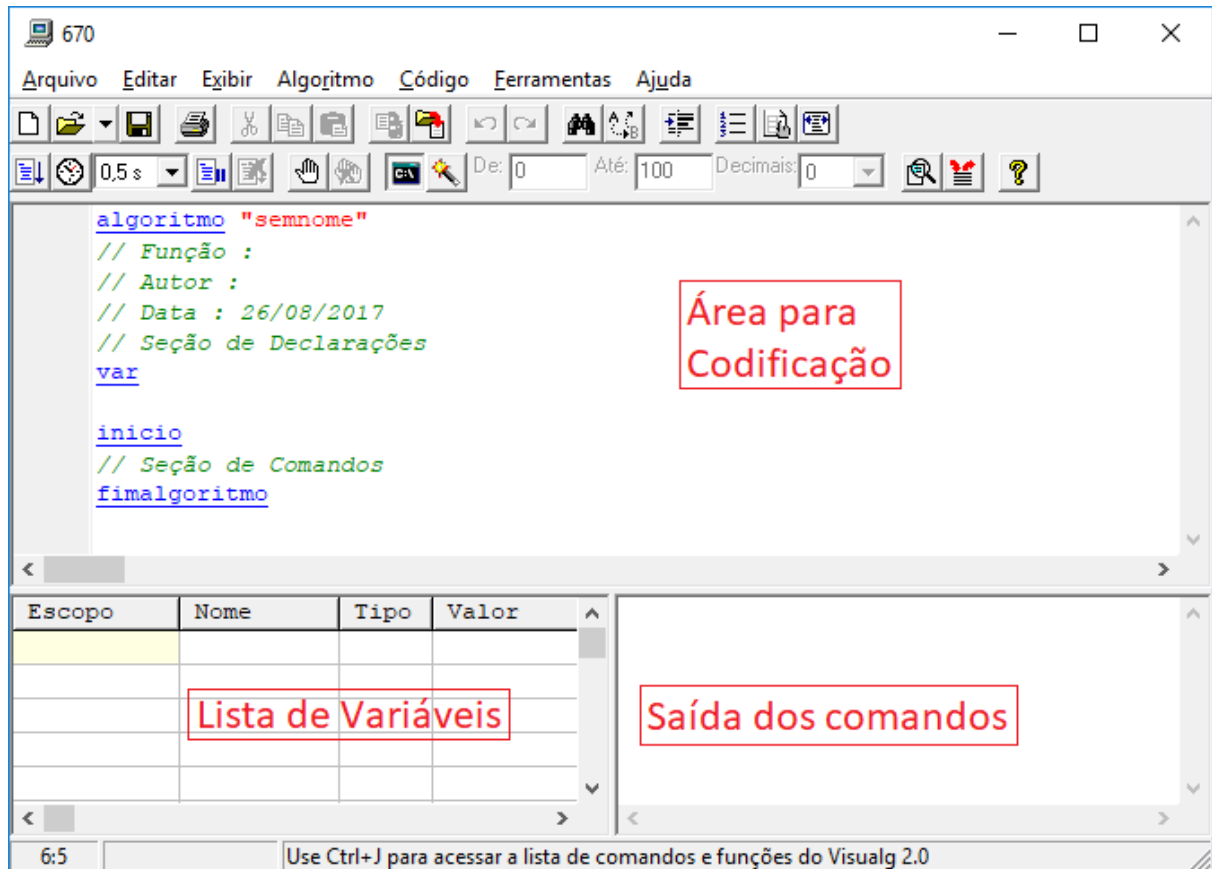
Fonte: Aguilar (2008, p. 37).

Concluída a parte introdutória, onde foi explanada a importância da utilização do diagrama de blocos, partiu-se para os trabalhos com o ambiente de pseudocódigo VisuAlg. A escolha deste ambiente pelo professor obedeceu a alguns requisitos listados abaixo:

- Ambiente atualizado e com linguagem nativa (língua portuguesa);
- Proximidade visual com a ferramenta de programação em linguagem C que é utilizada na sequência da disciplina;
- Ter uma interface clara e objetiva, como área de código com sintaxe colorida, listagem de variáveis e valores, execução e interrupção da execução do programa e execução passo a passo;
- Facilidade no *download* e instalação no próprio computador do estudante, fora da instituição, para poder praticar os exercícios em sua residência ou trabalho.

A figura a seguir mostra a tela de funcionamento do VisuAlg, com a marcação das principais áreas do programa.

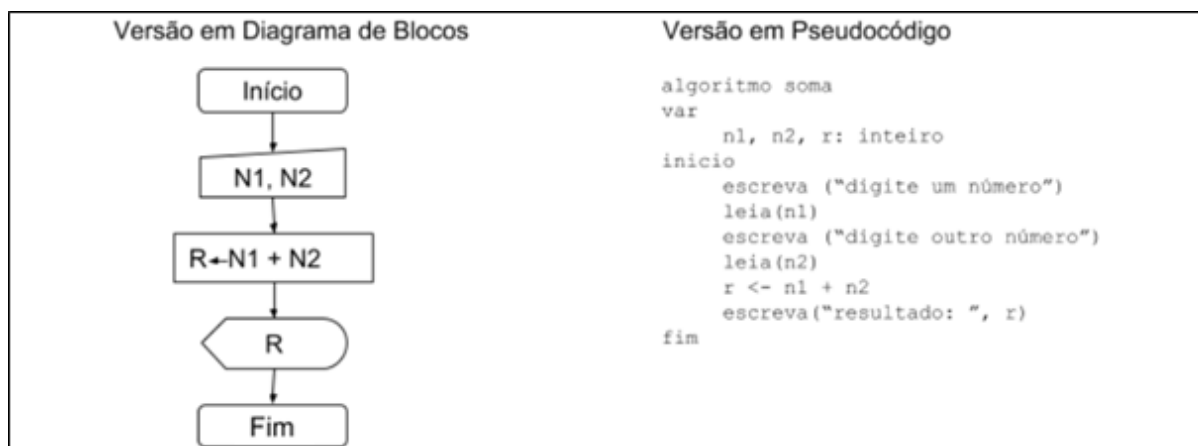
Figura 3 – Interface do programa VisuaAlg.



Fonte: Do autor (2017).

Durante a fase inicial da disciplina foram propostos aos estudantes vários exercícios com problemas de diferentes realidades para serem solucionados, alguns simples e outros mais complexos. Dentre os exercícios propostos, foi solicitado aos estudantes que criassem um Algoritmo, que receba dois números inteiros digitados pelo usuário e exibir a soma deles na tela. A Figura 4 mostra a elaboração deste Algoritmo, em diagrama de blocos e pseudocódigo.

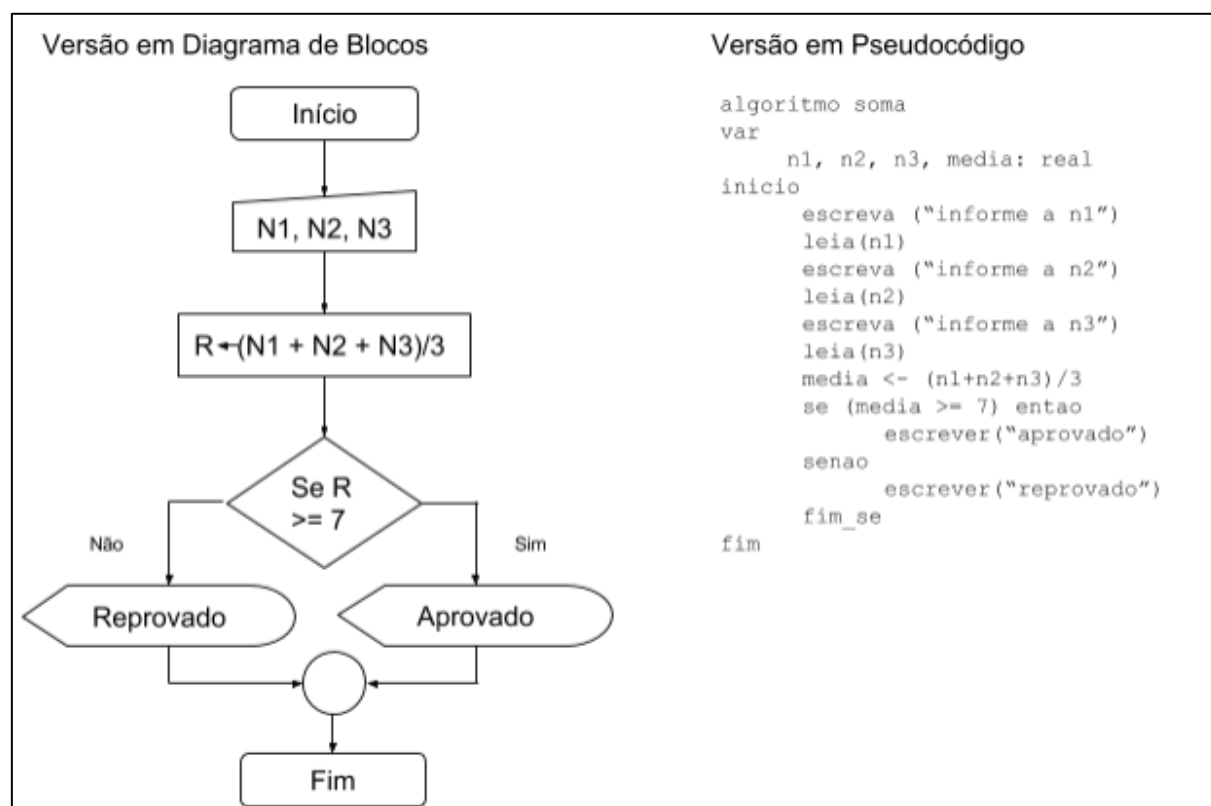
Figura 4 – Elaboração do Algoritmo em diagrama de blocos e pseudocódigo.



Fonte: Do autor (2017).

A Figura 5 ilustra outro Algoritmo, mais complexo que o anterior, em que o aluno é desafiado a criar uma solução que leia 3 notas de um estudante, calcule a média e em seguida informe se o mesmo está aprovado ou reprovado, considerando para isso a média 7.

Figura 5 – Algoritmo para calcular a média de um aluno.



Fonte: Do autor (2017).

A avaliação da aprendizagem sobre Algoritmos concentrou-se em analisar o conhecimento do estudante referente aos temas abaixo:

- Escrita de mensagens em tela - comando **escrever**;
- Leitura de dados pelo teclado com o comando **leia**;
- Cálculos com expressões aritméticas;
- Estruturas de seleção - comandos **se** e **escolha...caso**;
- Estruturas de repetição - comandos **para**, **enquanto** e **repita**.

É importante salientar a facilidade da leitura do código escrito em português estruturado. Através desta técnica o aprendizado de linguagens de programação se torna muito mais atrativo e compensador pois as instruções programadas são escritas em língua portuguesa. Na avaliação realizada, também se levaram em consideração outros critérios como criatividade, originalidade, proatividade, organização e clareza na escrita do código.

Cabe ressaltar que em semestres anteriores o conteúdo de Algoritmos não era ministrado na disciplina de Programação I. O contato dos alunos com estes tipos de exercício se dava diretamente em uma linguagem de programação real, de alto nível. Não haviam ilustrações e instruções em linguagem nativa, dificultando os processos de ensino e de aprendizagem.

Em muitos casos era visível a falta de conhecimento em língua inglesa, idioma original da maioria dos *softwares* de programação. O desafio se tornava duplo, pois, além da necessidade de se aprender a dominar a ferramenta e a linguagem, também necessitava-se compreender a construção de programas de computador.

Após concluídos os conteúdos relativos a Algoritmos, prosseguiu-se normalmente com a ementa da disciplina, como ocorrera em semestres anteriores. Alguns comportamentos positivos já puderam ser observados durante as aulas iniciais. Observou-se que o número de desistências foi bastante menor em relação a semestres anteriores além de que os estudantes questionavam em uma frequência muito maior o professor acerca do novo assunto.

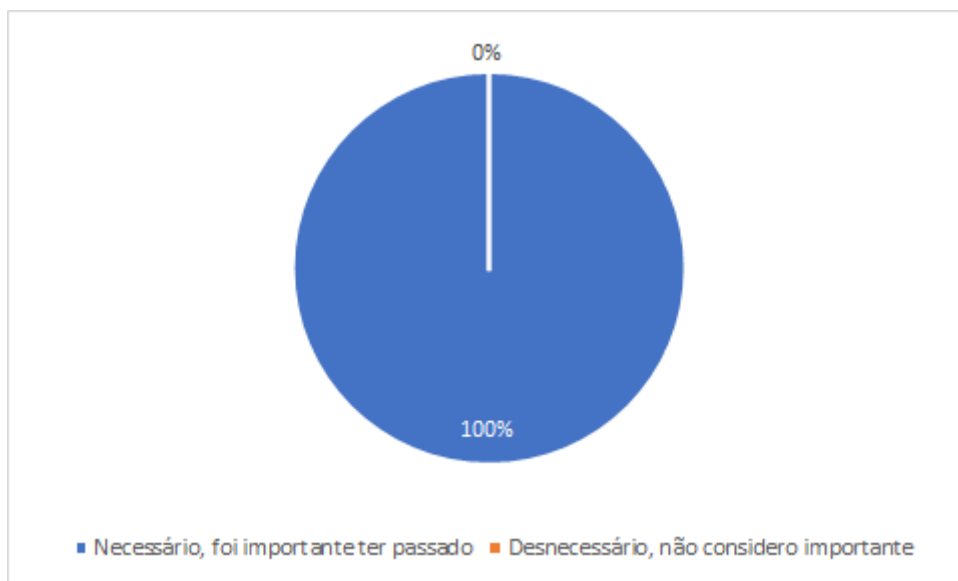
Pode-se concluir através desta realidade, um crescente interesse pelo conteúdo proposto. No final desta disciplina, observou-se um crescimento considerável na avaliação inicial dos alunos e, principalmente, uma estabilidade em relação à média final desta disciplina.

Seguem abaixo outros aspectos positivos observados na disciplina a partir da utilização de pseudocódigo no início da disciplina de Programação I:

- **Motivação:** observaram-se alunos muito mais motivados e comprometidos com a disciplina;
- **Transição:** diminuição no impacto da transição para a ferramenta de programação em alto nível;
- **Rapidez:** maior habilidade posterior na construção de programas em linguagem de alto nível;
- **Cooperação:** maior auxílio e troca de ideias entre os estudantes na busca pela solução dos problemas propostos.

No encerramento da disciplina de Programação I, que já teve o conteúdo de Algoritmos incorporado ao conteúdo programático, aplicou-se um questionário para avaliar junto aos alunos a importância dessa metodologia. Na oportunidade foram entrevistados 24 estudantes e todos responderam positivamente sobre a importância do novo conteúdo.

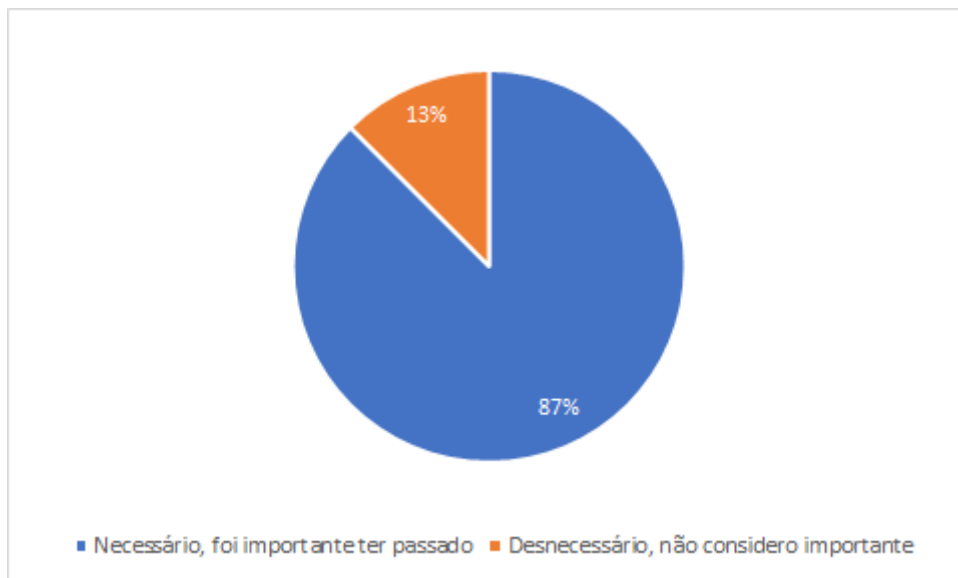
Figura 6 – Opinião dos alunos sobre a utilização de Algoritmos no conteúdo da disciplina.



Fonte: Do autor (2017).

Em outra pergunta deste mesmo questionário, foi solicitado aos estudantes se eles consideravam importante aprender programação previamente em um ambiente de pseudocódigo ou se deveriam evoluir diretamente para a linguagem de programação de alto nível. Como resultado, a ampla maioria (correspondente a 87% dos entrevistados) considerou importante ter passado este conteúdo.

Figura 7 - Opinião dos alunos sobre a utilização do pseudocódigo.



Fonte: Do autor (2017).

Entretanto, 13% dos alunos entenderam que não era necessário desenvolver o tema de Algoritmos. Acredita-se que estes estudantes já possuem algum entendimento prévio do assunto e conseqüentemente uma facilidade maior com o conteúdo proposto.

Pretende-se no futuro evoluir para uma pesquisa mais aprofundada acerca desse tema. É importante explorar critérios mais concretos para monitorar a evolução dos estudantes com esta adaptação no conteúdo programático, já que temos realidades muito distintas de cada discente no cotidiano da sala de aula. Também realizar experiências com outras ferramentas, tais como as levantadas anteriormente, para analisar em qual delas os estudantes possuem um melhor desempenho na aprendizagem.

5 CONCLUSÃO

O ensino de linguagens de programação sempre foi um grande desafio, tanto para professores, quanto alunos. Pelo fato de ser um conteúdo com um nível de dificuldade considerável, cabe ao educador encontrar alternativas e metodologias para facilitar os processos de ensino e de aprendizagem em sala de aula, minimizando as dificuldades que surgem.

O objetivo deste trabalho foi efetuar um levantamento da necessidade e importância em se ministrar o conteúdo de Algoritmos na disciplina de Programação I, do curso técnico em Eletroeletrônica, da Univates.

Tomando como base os semestres monitorados, após a inserção do novo conteúdo na disciplina, nas cinco primeiras aulas, registrou-se um crescimento satisfatório no nível de aprendizado em programação de computadores. O desinteresse pelas aulas já não era mais visível e a evasão diminuiu bastante, comparado a semestres anteriores, comprovando que houve um aumento de interesse pelo conteúdo.

Para consultar a opinião dos discentes sobre esta nova proposta de conteúdo programático, aplicou-se um breve questionário onde os mesmos ratificaram a importância de se ministrar Algoritmos como um aliado nos processos de ensino e de aprendizagem de programação de computadores.

De acordo com os resultados obtidos, constatou-se que quase 90% dos alunos entrevistados concordaram com a modificação realizada na disciplina de Programação I. Espera-se repetir esta experiência nos próximos semestres, realizando um monitoramento mais apurado dos seus resultados e mensurando novas variáveis.

REFERÊNCIAS

AGUILAR, Luis Joyanes. **Fundamentos de Programação**: Algoritmos, estruturas de dados e objetos. 3. ed. São Paulo: AMGH, 2011.

FALCKEMBACH, Gilse A. Morgental; ARAÚJO, Fabrício Viero de. **Aprendizagem de Algoritmos**: Dificuldades na Resolução de Problemas. Disponível em <<http://periodicos.unesc.net/sulcomp/article/view/916/909>>. Acesso em 31 de ago de 2017.

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. **Lógica de Programação**: A construção de Algoritmos e estruturas de dados. 3. ed. São Paulo: Prentice Hall, 2005.

GOMES, Anabela et al. **Aprendizagem de programação de computadores**: dificuldades e ferramentas de suporte. Revista Portuguesa de Pedagogia, [S.l.], p. p. 161-179, jul. 2008. Disponível em: <<http://impactum-journals.uc.pt/rppedagogia/article/view/1242>>. Acesso em 1 de set de 2017.

GOODRICH, Michael T.; TAMASSIA, Roberto. **Projeto de Algoritmos**: fundamentos, análise e exemplos da Internet. Porto Alegre: Bookman, 2008.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. **Algoritmos**: Lógica para Desenvolvimento de Programação de Computadores. 28. ed. São Paulo: Érica, 2016.

MANZANO, José Augusto N. G. et al. **Algoritmos: Técnicas de Programação**. 2. ed. São Paulo: Érica, 2015.

SILVA, Bruno Siqueira da; TRENTIN, Marco Antônio Sandini. **Dificuldades no Ensino-Aprendizagem de Programação de Computadores**: Contribuições para sua Compreensão e Resolução. Disponível em: <<http://www.sinect.com.br/2016/down.php?id=3349&q=1>>. Acesso em 1 de set de 2017.